UNITED STATES PATENT APPLICATION
FOR

SINGLE ARRAY BANKED BRANCH TARGET BUFFER

INVENTOR:

NICOLAS I. KACEVAS

PREPARED BY:

KENYON & KENYON
1500 K ST., N.W.
WASHINGTON, D.C. 20005

(202) 220-4200

SINGLE ARRAY BANKED BRANCH TARGET BUFFER

BACKGROUND

The present invention relates to processors. More particularly, the present invention relates to a single array banked branch target buffer for use in a processor.

5   Many processors, such as a microprocessor found in a computer, use an instruction pipeline to speed the processing of an instruction stream. The pipeline has multiple stages and an instruction is processed using early pipeline stages before a previous instruction is executed by later stages. Some instructions, such as a "conditional branch" instruction, however, must be executed before the processor can determine which instruction should be processed next. To

10   increase efficiency, the processor "predicts" when a conditional branch is being processed and also predicts which instruction, or "branch," should be placed into the early pipeline stages before that conditional branch instruction is executed.

To predict branch instructions within the instruction stream, a cache called a "branch target buffer" can be used to store information about branch instructions previously executed by

15   the processor.

An instruction fetch unit may fetch upcoming instructions by fetching several bytes at the same time, such as by fetching sixteen-byte blocks of memory. A single block, however, may contain multiple branch instructions. In such a processor, the branch prediction mechanism must find the first taken branch instruction, if any, within the current sixteen-byte block to determine

20   the next sixteen-byte block to fetch.

To efficiently find the first taken branch instruction, a "banked" branch target buffer may be used. U.S. Patent No. 5,842,008 entitled "Method and Apparatus for Implementing a Branch Target Buffer Cache with Multiple BTB Banks" discloses such a banked branch target buffer.

13107

Each bank in the branch target buffer contains branch entries for branch instructions located in a different subset of the memory blocks. The branch target buffer banks are ordered such that when a branch prediction must be made, the branch entry "hits" produced by each branch target buffer bank are ordered, reducing the amount of time required to prioritize the branch

5   instructions.

The banked branch target buffer uses a different, independent array for each bank, such that each array provides information about branches residing in the associated bank. The information is later prioritized. By way of example, a branch target buffer can use four banks to store information about a sixteen byte cache line. The branch target buffer may store information

10   about a single branch in each bank, and therefore a maximum of four branches would be stored for each sixteen byte cache line.

FIG. 1 illustrates a known architecture for a single bank in such a banked branch target buffer. In other words, a branch target buffer having four banks would need four of the circuits shown in FIG. 1 (one for each bank). For each bank, a lookup IP (Instruction Pointer) signal

15   passes through a multiplexer 100 coupled to four buffer entries 200, or "ways." Note that the number of ways (four) does not have to be the same as the number of banks in the branch target buffer. The buffer entries 200 are mapped to cache data buffers 300, and each buffer entry 200 is coupled to a comparator 400. The lookup IP signal is also input to the comparators 400, and the results from each comparator 400 are coupled to another multiplexer 500 that selects cache data

20   stored in the cache data buffers 300.

Each bank of the branch target buffer is addressed by the sixteen-byte line address. When a lookup to the branch target buffer occurs, each bank is addressed simultaneously and provides

13107   2

information for that particular bank and line. The result of each bank's search is then combined by a prioritizing and merging logic unit to produce the final branch target buffer prediction.

Assuming that each bank is organized in a classic four-way set associative fashion, a method for performing a look-up is illustrated in FIG. 2 for one of the four branch target buffer

5   banks. The lookup address, or Instruction Pointer (IP) is split into an "IP tag" field and an "IP set" field at step 10. The IP set field corresponds to a specific bank in the branch target buffer. At step 20, the IP set field is decoded and used to read out the four ways corresponding to that set (i.e., the four entries in the specific branch target buffer bank). The tag, valid and data fields are then read for each of the four ways or entries at step 30. These four entry tags are compared to

10   the IP tag in order to determine from which valid way, if any, there is a match at step 40, and the chosen way is then used to select the data that represents that particular bank's information.

In a traditional four-way set associative organization (such as the one shown in FIG.1), it is forbidden to have the same valid tag in different ways of the same set. In other words, the match signals from the comparators and valid bit combined are mutually exclusive.

15   Providing independent arrays for each bank, however, creates routing problems, and increases area and design overhead related to the banked branch target buffer implementation. In particular, the use of multiple arrays implies costly routing from each array to the merging logic, which translates directly into timing problems.

Moreover, the utilization of the available entries is limited when multiple arrays are used.

20   For example, branches may not be uniformly distributed between the banks. Suppose that most branches sit in bank 0. In this case, there may be a lot of replacements in bank 0 while entries in banks 1 to 3 are less utilized.

SUMMARY

In accordance with an embodiment of the present invention, an Instruction Pointer (IP) signal is received comprising an IP tag field and an IP set field. A plurality of entries corresponding to the IP set field are read, each of the entries comprising an entry tag, an entry

5   bank, and entry data. Each entry tag and entry bank is then compared with the IP tag and each of the plurality of banks.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 illustrates a known architecture for a single bank in a banked branch target buffer.

10   FIG. 2 is a flow diagram of a known method for performing a look-up using a banked branch target buffer.

FIG. 3 is a banked branch target buffer according to an embodiment of the present invention.

FIG. 4 is a flow diagram of a method for using a banked branch target buffer according to

15   an embodiment of the present invention.

FIG. 5 is a branch prediction circuit according to an embodiment of the present invention.

DETAILED DESCRIPTION

An embodiment of the present invention is directed to a single array banked branch target buffer. Referring now in detail to the drawings wherein like parts are designated by like reference numerals throughout, FIG. 3 illustrates a four-bank branch target buffer according to an embodiment of the present invention. A lookup IP signal passes through a multiplexer 150 coupled to four buffer entries 250, or "ways." Note that the number of ways (four) does not have to be the same as the number of banks in the branch target buffer. The buffer entries 250 are mapped to cache data buffers 350, and each buffer entry 250 is coupled to four comparators 450. The lookup IP signal is also input to the comparators 450 together with a hard-wired bank number (i.e., "00," "01," "10" and "11"), and the results from each comparator 400 are coupled to multiplexers 550 that select cache data stored in the cache data buffers 350. As a result, information for four banks is selected.

Thus, a banked branch target buffer is provided having a single array, as opposed to having an array for every bank. The array has an extra field, next to the tag, which specifies which bank an entry belongs to. Note that - by way of example only - the unique array can be four-way set associative, although the number of ways is not related to the number of banks. It will also be understood by those skilled in the art that a number of banks other than four banks may be used instead.

FIG. 4 is a flow diagram of a method for using a banked branch target buffer according to an embodiment of the present invention. At step 15, the lookup address, or Instruction Pointer (IP) is split into an "IP tag" field and an "IP set" field. The IP set field is decoded and used to read out the four ways (entries) corresponding to that set (bank) at step 25. The tag, valid and data fields are then read for each of the four ways at step 35.

13107

5

The four entry tags concatenated with the bank number are then compared to the IP tag, concatenated also with all the possible bank numbers, at step 45. This means that for every possible bank number there is a set of four (i.e., the number of ways) comparisons. In the example shown in FIG. 3, there are four banks and four ways, so the total number of comparisons is 16. Each of the sets of four comparators provide four results, whose meaning is that there is a match for that bank and for that particular way.

Each chosen way for a bank set of comparators is then used to select the particular bank's information. In this way, up to four banks of branch target buffer information may be obtained from a single branch target buffer array.

According to an embodiment of the present invention, the valid tag and bank in different ways of the same set may be unique. In other words, the match signals from the same bank's comparators and valid bit combined should be mutually exclusive.

Note that although from a logic point of view it seems that there is a lot of overhead by having four sets of four comparators, in practice most of the logic of the comparators for a given way can be shared. This is because the compared values are almost identical. The different bits are constants from one of the sources, which may enable further optimizations.

FIG. 5 is a branch prediction circuit 600 according to an embodiment of the present invention. The Instruction Pointer (IP) is received by a branch target buffer (BTB) cache 610 that has a number (n) of banks arranged as a single array. A prioritizer circuit 620 uses information from the BTB cache 610 to determine a result.

The branch prediction circuit 600 may be used, for example, in a computer processor coupled to a memory, the memory being memory divided into memory blocks. In this case, the branch prediction circuit 600 would predict a block of memory to fetch based upon the IP (that

points to a currently executing instruction). The BTB cache 610 shown in FIG. 5 comprises a plurality of ordered branch target buffer banks (formed as a single array), comprising a plurality of branch entries storing information about branch instructions addressed by address bits specifying a different subblock within said memory blocks. The branch prediction circuit 600

5   receives IP and indexes into all of the ordered branch target buffer banks of the BTB cache and fetches at most one branch entry from each bank. The prioritizer circuit 620 indicates the selection of one of the branch entries fetched by the branch prediction circuit 600 from the ordered branch target buffer banks by selecting a first taken branch instruction located after the IP.

10   Providing a single array for all of the banks reduces routing problems as well as area and design overhead. Moreover, the utilization of the available entries is improved even when branches are not uniformly distributed between the banks. Suppose that most branches sit in bank 0. In this case, there will merely be more instances of bank 0 throughout the array.

            Although various embodiments are specifically illustrated and described herein, it will be

15   appreciated that modifications and variations of the present invention are covered by the above teachings and within the purview of the appended claims without departing from the spirit and intended scope of the invention. Moreover, the present invention applies to a broad range of banked branch target buffer architectures, and is therefore a general approach that includes a broad range of specific implementations. In addition, although software or hardware are

20   described to control certain functions, such functions can be performed using either software, hardware or a combination of software and hardware, as is well known in the art. As is also known, software may be stored, such as in memory, in the form of instructions, including micro-code instructions, adapted to be executed by a processor. As used herein, the phrase "adapted to

be executed by a processor" encompasses instructions that need to be translated before being

executed by the processor.